User Experience Implementing SSL and Terminal Servers in z/VM 6.1

Jim Moling
US Treasury, Financial Management Service

Friday, August 12, 2011

Session Number **10047**

Disclaimers

- **The opinions & ideas expressed herein are those of the author alone and do not necessarily reflect those of Financial Management Service, furthermore, Financial Management Service is hereby absolved of any and all responsibility or liability for the information contained herein.**
- **Copyrights & Trademarks:**
  - Any and all copyrights & trademarks are hereby acknowledged to be owned by their respective parties
  - All other brands, logos and products are trademarks or registered trademarks of their respective companies
  - All rights reserved
- **Disclaimer of Endorsement:**

  Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government, and shall not be used for advertising or product endorsement purposes.
- **Disclaimer of Liability:**

  With respect to this presentation, neither the United States Government nor any of their employees, makes any warranty, express or implied, including the warranties of merchantability and fitness for a particular purpose, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Introduction

- This presentation describes how a user implemented the SSL TCP/IP server for secure access to z/VM (version 6.1) as well as how the new Terminal Server has been implemented for access to virtual Linux servers. The goal of this implementation was to achieve a more secure and centralized means of accessing VM and the Linux servers running under it. This session will show a before and after configuration and the steps taken to achieve the stated goals in a step-by-step how-to fashion.
- Another incentive (perhaps even the driving force) behind accomplishing these goals was to satisfy audit findings

Overview

Part 1 - Implementing an SSL Server on z/VM 6.1

- What is an SSL Server? Why do we want to use it?
- What is needed for implementation
- Steps for basic implementation
  - Overview of steps
  - Recipe
- Next steps

Part 2 - Implementing a Terminal Server on z/VM 6.1

- What is a Terminal Server? Why do we want to use it?
- What is needed for implementation
- Steps for basic implementation
  - Overview of steps
  - Recipe
- Next steps

Overview (Continued)

- Summary
- Questions

Part 1 - Implementing an SSL Server

What is the (CMS-based) SSL Server for z/VM?

- SSL (Secure Sockets Layer) was developed to provide point-to-point encryption of TCP/IP traffic
- Standardized by RFC 2246 as TLS (Transport Layer Security)
- Provides security in a z/VM environment for any server associated with a TCP/IP stack
- Prior to z/VM 5.4, only a Linux-based SSL Server option was available
- A CMS-based SSL Server was introduced with z/VM 5.4
- It's a component of TCP/IP that ships with z/VM 6.1

Part 1 - Implementing an SSL Server

Why do we want to use it?

- Allows us to access VM via a Telnet client, such as IBM Personal Communications, in a secure fashion
- Allows us to perform FTP-based file transfers in a secure fashion

- SSL Server references that this presentation is based on:
  - http://www.vm.ibm.com/related/tcpip/tcsslspe.html
  - Presentation: **z/VM SSL Server Update** by Brian Hugenbruch

Part 1 - Implementing an SSL Server

What is needed for implementation

- Install z/VM 6.1 + configure with basic TCP/IP access
  - Either follow instructions from IBM that come with z/VM 6.1, or
  - Use the Virtualization Cookbook for SLES11 (Redbook SG24-7931-00) to install z/VM 6.1
- Make sure that PTF UK59536 is applied
  - Provides required updates for the SSL & TCP/IP components
- The implementation that follows is based on a z/VM 6.1 system after the initial installation of the shipped configuration and then applying PTF UK59536
- Assumes the default VM System ID is used – ZVMV6R10

Part 1 - Implementing an SSL Server

Overview of steps to implement an SSL Server

- Determine the SSL Server Configuration For Your Installation
- Update the TCP/IP server configuration file (PROFILE TCPIP)
- Update the DTCPARMS file for the TCP/IP server
- Update the DTCPARMS file for the SSL Server and the (new) DCSS Management Agent server
- Update the DTCPARMS file for the SSL Server Daemon
- Set up the Certificate Database
- Bounce TCPIP to start up the SSL Server

Part 1 - Implementing an SSL Server

Determine the SSL Server Configuration For Your Installation

- Secure communications support can be provided via one of the following SSL configurations:
  - A single-instance SSL server, or
  - A server "pool," for which multiple SSL servers are employed
- We are choosing to implement the single-instance SSL server option to keep it simple.

Part 1 - Implementing an SSL Server

Update the TCP/IP Server Configuration File (PROFILE TCPIP)

- Logon to TCPMAINT and do the following:
- Copy PROFILE TCPIP D1 to ZVMV6R10 TCPIP D1
  - Command: COPYFILE PROFILE TCPIP D ZVMV6R10 = D
- Note: 'ZVMV6R10' is the system name. When TCPIP is started, it looks for file names = system name first, i.e. sysname TCPIP, sysname DTCPARMS, etc.
- Xedit ZVMV6R10 TCPIP D and add the following SSL Server related statements:
  - SSLSERVERID SSLSERV TIMEOUT 60
  - SSLLIMITS MAXSESSIONS 1000 MAXPERSSLSERVER 100

Part 1 - Implementing an SSL Server

Update the DTCPARMS File for the TCP/IP Server

- Include a :DCSS_Parms. tag for the TCP/IP server with which the SSL server is to provide secure communications support.
- Copy SYSTEM DTCPARMS D1 to ZVMV6R10 DTCPARMS D1
  - Command: COPYFILE SYSTEM DTCPARMS D ZVMV6R10 = D
- Xedit ZVMV6R10 DTCPARMS D and update the TCPIP server definition

Part 1 - Implementing an SSL Server

Update the DTCPARMS File for the TCP/IP Server

- Xedit ZVMV6R10 DTCPARMS D and update the TCPIP server definition:

```
:nick.TCPIP      :type.server
                 :class.stack
                 :attach.1130-1132
                 :DCSS_Parms.<DEFAULT>
```

Part 1 - Implementing an SSL Server

Update the DTCPARMS file for the SSL Server and the DCSS Management Agent server

- Add the following definitions:

```
:nick.SSLSERV   :type.server
                :class.ssl
                :stack.TCPIP
.*
:nick.SSLDCSSM  :type.server
                :class.ssl_dcss_agent
                :stack.TCPIP
                :for.SSLSERV
```

Part 1 - Implementing an SSL Server

Update the DTCPARMS file for the SSL Server Daemon

- Add the following definition:

```
.* Secure Socket Layer (SSL) daemon
:nick.ssl       :type.class
                :name.SSL daemon
                :command.VMSSL
                :runtime.C
                :diskwarn.YES
                :Admin_ID_list.TCPMAINT GSKADMIN
                :memory.256M
                :mixedcaseparms.YES
                :mount. /../VMBFS:VMSYS:ROOT/       /      ,
                        /../VMBFS:VMSYS:SSLSERV/    /tmp   ,
                        /../VMBFS:VMSYS:GSKSSLDB/   /etc/gskadm
                :parms.KEYFile /etc/gskadm/TstCerts.kdb
```

- Logoff TCPMAINT

Part 1 - Implementing an SSL Server

Setup the Certificate Database

- Log on the **GSKADMIN** user ID and allow its default PROFILE EXEC to run
- Invoke the **gskkyman** utility. A menu is displayed:

```
Database Menu

1 - Create new database
2 - Open database
3 - Change database password
4 - Change database record length
5 - Delete database
6 - Create key parameter file
7 - Display certificate file (Binary or Base64 ASN.1 DER)

0 - Exit program
```

Part 1 - Implementing an SSL Server

Setup the Certificate Database

Select option 1 – Create new database, and then respond to the following prompts:

```
Enter key database name (press ENTER to return to menu):
        TstCerts.kdb
Enter database password (press ENTER to return to menu):
        tstadmin
Re-enter database password:   tstadmin
Enter password expiration in days (press ENTER for no expiration):
        <Enter>
Enter database record length (press ENTER to use 5000):     <Enter>

Key database /etc/gskadm/TstCerts.kdb created.
```

Part 1 - Implementing an SSL Server

Setup the Certificate Database

- Select option 10 - Store database password (you should receive the following reply):

```
Database password stored in /etc/gskadm/TSTCERTS.sth
```

- Exit the **gskkyman** program by selecting option 0.
- Issue the **OPENVM** commands that follow to confirm that the necessary database files have been created and to list the permissions of these files.

Part 1 - Implementing an SSL Server

OPENVM Commands

- **openvm list /etc/gskadm/**

```
Directory = '/etc/gskadm/'
Update-Dt  Update-Tm Type  Links          Bytes Path name component
07/31/2011 19:08:44   F        1          60080 'TstCerts.kdb'
07/31/2011 19:12:57   F        1             80 'TstCerts.rdb'
07/31/2011 19:11:48   F        1            129 'TstCerts.sth'
```

- **openvm list /etc/gskadm/ (own**

```
Directory = '/etc/gskadm/'
User ID    Group Name  Permissions Type  Path name component
gskadmin   security    rw- --- ---  F    'TstCerts.kdb'
gskadmin   security    rw- --- ---  F    'TstCerts.rdb'
gskadmin   security    rw- --- ---  F    'TstCerts.sth'
```

Part 1 - Implementing an SSL Server

OPENVM Commands

- Issue the **OPENVM PERMIT** commands that follow to allow the SSL server to access the newly-created key database:

```
openvm permit /etc/gskadm/TstCerts.kdb rw- r-- ---
openvm permit /etc/gskadm/TstCerts.sth rw- r-- ---
```

- Issue the **OPENVM LIST** command that follows to confirm that r (read) has been added to the "group" permissions for the key database and password stash files:

```
openvm list /etc/gskadm/ (own
Directory = '/etc/gskadm/'
User ID    Group Name  Permissions Type  Path name component
gskadmin   security    rw- r-- ---   F    'TstCerts.kdb'
gskadmin   security    rw- --- ---   F    'TstCerts.rdb'
gskadmin   security    rw- r-- ---   F    'TstCerts.sth'
```

- **Logoff GSKADMIN**

Part 1 - Implementing an SSL Server

- With the key database now in place, the SSL server can be initialized to confirm it has access to this database.
- Bounce TCPIP and see if the SSL Server starts:
  - To shutdown TCPIP: `FORCE TCPIP`
  - To restart TCPIP: `XAUTOLOG TCPIP`
- Issue the '`Query Names`' command to confirm that SSLSERV & SSLDCSSM are active.
- The key database can now be populated with the appropriate server and CA certificates required to provide SSL-protected communications for your installation. For more information, see z/VM: *TCP/IP Planning and Customization* and *TCP/IP User's Guide* manuals.

Part 1 - Implementing an SSL Server

Next Steps

- SSL can now be used to setup secure access to VM via a Telnet client, such as IBM Personal Communications
- SSL can now be used to perform FTP-based file transfers in a secure fashion

What is a Terminal Server?

- A *terminal server* is a Linux instance that provides access to terminal devices on other Linux instances, called *target systems*.
- The terminal server and all target systems run as guest operating systems of the same z/VM instance.
- Terminal server and target systems are connected through the z/VM Inter-User Communication Vehicle (IUCV).

Part 2 - Implementing a Terminal Server

Why do we want to use it?

- From the terminal server, administrators can access terminal devices on target systems without requiring direct TCP/IP connections to the target systems.
- You can use a terminal server to:
  - Increase availability by providing emergency access to target systems if the primary network for these systems fails.
  - Heighten security by separating user networks from administrator networks or by isolating sensitive Linux instances from IP networks.
  - Simplify systems administration by providing a central access point to target systems.

Part 2 - Implementing a Terminal Server

Terminal Server references this presentation is based on

- *How to Set up a Terminal Server Environment on z/VM* (SC34-2596-00)
- *Device Drivers, Features, and Commands* (SC33-8411-11)
- *The Virtualization Cookbook for SLES 11 SP1* (SG24-7931-00)

Part 2 - Implementing a Terminal Server

What is needed for implementation

- This presentation is based on SUSE Linux Enterprise Server Version 11 Service Pack 1 (SLES11 SP1)
- Use the *Virtualization Cookbook for SLES11* (Redbook SG24-7931-00) to create a cloning server
- 2 servers are created from the cookbook:
  - S11S1CLN, the cloning server
  - S11S1GLD, the Golden Image server
- 2 servers are then cloned for Terminal Server:
  - TRMSRV, a cloned server for use as the Terminal Server
  - TSTSRV, a client that is used for testing Terminal Server

Part 2 - Implementing a Terminal Server

Overview of Steps to Implement a Terminal Server

- Terminal Server code already included in SLES11 SP1
- Update VM Directory for IUCV Access
  - Create a separate profile for Terminal Server - TRMSRV
  - Update profiles for IUCV access
- Setup the TRMSRV server as a Terminal Server
  - Define a user ID for testing
  - Define access authorizations
- Setup the TSTSRV server as a Terminal Server Client
  - Define a user ID for testing
  - Define terminal types

Part 2 - Implementing a Terminal Server

Terminal Server code already included in SLES11 SP1

- Most of the work involves minor updates, as the Terminal Server code is contained in the s390-tools package, which is pre-installed in SLES11 SP1.
- S390-tools package version 1.8.1 or later is required.
- We will focus on basic functionality, however, there are several additional features that can be exploited, such as session logging and the ability to replay a session.

Part 2 - Implementing a Terminal Server

Update the VM Directory for IUCV Access

- Logon to MAINT and xedit the User Direct file
- The LNXDFLT profile entry is created via the cookbook
- Make a copy of the LNXDFLT profile entry and call it TERMSERV
- Add the IUCV ANY statement to the LNXDFLT profile

```
PROFILE LNXDFLT
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  IUCV ANY
  ...
```

Part 2 - Implementing a Terminal Server

Update the VM Directory for IUCV Access

- Add the IUCV ANY and MAXCONN statements to the TERMSERV profile (copied from LNXDFLT)

```
PROFILE TERMSERV
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  IUCV ANY
  OPTION MAXCONN 128
  ...
```

Part 2 - Implementing a Terminal Server

Update the VM Directory for IUCV Access

- Change the profile for the TRMSRV entry to TERMSERV, so it is similar to the following:

```
USER TRMSRV   NEWSYS   512M  1G G
  INCLUDE TERMSERV
  OPTION APPLMON
  MDISK 100 3390 00001 03338 LNX062 MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 03339 03338 LNX062 MR LNX4VM LNX4VM LNX4VM
```

Part 2 - Implementing a Terminal Server

Setup the TRMSRV server as a Terminal Server

- Login to TRMSRV as root
- Define a userid – jmoling

```
useradd -s /usr/bin/ts-shell -G ts-shell jmoling
```

- Define a home path

```
mkdir /home/jmoling
```

- Define Authorizations

```
cd /etc/iucvterm/
vi ts-authorization.conf
```

Add: `jmoling=list:mebmon`

Save changes: `:wq`

- Bounce TRMSRV so changes take affect: `Reboot`

Part 2 - Implementing a Terminal Server

Setup the TSTSRV server as a Terminal Client

- Login to TSTSRV as root
- Define a userid – jmoling

```
useradd -s /usr/bin/ts-shell -G ts-shell jmoling
```

- Define a home path

```
mkdir /home/jmoling
```

- Edit inittab and add a terminal definition

```
Cd /etc
Vi zipl.conf
```

Add: `i1:2345:respawn:/usr/bin/iucvtty lxterm1`

Save changes: `:wq`

Part 2 - Implementing a Terminal Server

Setup the TSTSRV server as a Terminal Client

- Edit zipl.conf:

```
vi /etc/zipl.conf
```

- Add kernel parameters:

```
hvc_iucv=2 console=hvc0 consloe=ttyS0 hvc_iucv_allow=mebmon,jmoling
```

- Save changes:

```
:wq
```

- Update zipl

```
mkinitrd
zipl
```

- Bounce TSTSRV so changes take affect: `Reboot`

Part 2 - Implementing a Terminal Server

Next Steps

1. Logging onto the TRMSRV Terminal Server

```
login as: jmoling
Using keyboard-interactive authentication.
Password: ........
Last login: Fri Aug 12 03:15:36 2011 from …
Welcome to the Terminal Server shell.
Type 'help' to get a list of available commands.
```

Part 2 - Implementing a Terminal Server

Next Steps

2. Displaying the Help command

```
jmoling@ts-shell> help
Terminal Server shell help

Available commands:
        list                    List authorizations.
        connect <vm_guest>      Connect to specified z/VM guest virtual machine.
        terminal [<identifier>] Display or set the terminal identifier.
        q | quit | exit         Exit the current shell session.
        help                    Display help information.
        version                 Display version information.
```

Part 2 - Implementing a Terminal Server

Next Steps

3. Connecting to the TSTSRV client server

```
jmoling@ts-shell> connect tstsrv lxterm1
ts-shell: Connecting to tstsrv (terminal identifier: lxterm1)...

TSTSRV login: jmoling
Password:
Last login: Fri Aug 12 03:37:05 CDT 2011 from MEBTRM on pts/0
Directory: /home/jmoling
Sat Aug 13 00:36:35 CDT 2011
jmoling@MEBMON:~>
```

Summary

SSL Server

- SSL is now available to setup secure access to VM via a Telnet client, such as IBM Personal Communications
- SSL is now available to setup FTP-based file transfers in a secure fashion

Terminal Server

- The Terminal Server can now be used to connect to other servers that have been setup as a client

Thank You For Attending!

- Questions?

- Session **10047**